**SQL Server Naming Standards**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | 2013 |  |
|  |  |  |
|  |  |  |

.................................................................................................................................. 4

1.1. Instances ............................................................................................................................................. 4

1.2. Databases ............................................................................................................................................ 4

1.2.1. All Objects .................................................................................................................................... 4

1.2.2. General Naming ........................................................................................................................... 5

1.2.3. Database ...................................................................................................................................... 5

1.2.4. Filegroups..................................................................................................................................... 6

1.2.5. Database Files .............................................................................................................................. 6

1.2.6. Schema ......................................................................................................................................... 6

1.2.7. Table ............................................................................................................................................ 7

1.2.7.1. Data Warehouse and Data Mart........................................................................................... 7

1.2.8. Column ......................................................................................................................................... 7

1.2.8.1. System Generated IDs .......................................................................................................... 7

1.2.8.2. Boolean Columns .................................................................................................................. 8

1.2.8.3. Bitmask columns ................................................................................................................... 8

1.2.8.4. Date and Time Columns ........................................................................................................ 8

1.2.9. View ............................................................................................................................................. 8

1.2.10. User Defined Data Type ............................................................................................................. 8

1.2.11. Foreign key................................................................................................................................. 9

1.2.12. Primary key ................................................................................................................................ 9

1.2.13. Index .......................................................................................................................................... 9

1.2.14. Default ..................................................................................................................................... 10

1.2.15. Check constraint ...................................................................................................................... 10

1.2.16. Stored procedure ..................................................................................................................... 10

1.2.17. Trigger ...................................................................................................................................... 11

1.2.18. Function ................................................................................................................................... 11

1.2.19. XML Schema Collection ........................................................................................................... 11 .................................................................................................................... 11

1.2.21. Partition Scheme ..................................................................................................................... 12

1.2.22. Service Broker .......................................................................................................................... 12

1.2.22.1. Message Type ................................................................................................................... 12

1.2.22.2. Contract ............................................................................................................................ 12

1.2.22.3. Queue ............................................................................................................................... 12

1.2.22.4. Service ............................................................................................................................... 12

1.2.22.5. Endpoint............................................................................................................................ 13

1.2.22.6. Route ................................................................................................................................. 13

1.2.23. Linked Server ........................................................................................................................... 13

1.2.23.1. Replication ........................................................................................................................ 13

1.2.23.1.1. Publication ................................................................................................................. 13

1.2.23.1.2. Subscription ............................................................................................................... 13

1.2.23.2. SQL Server Agent .............................................................................................................. 14

1.2.23.3. Audit ................................................................................................................................. 14

1.2.23.4. Policy Management .......................................................................................................... 14

1.2.23.4.1. Policy .......................................................................................................................... 14

1.2.23.4.2. Condition ................................................................................................................... 14

1.2.23.5. Extended Event Session .................................................................................................... 14

1.3. SSIS .................................................................................................................................................... 14

1.3.1. Package Name ........................................................................................................................... 14

1.3.2. Configuration File....................................................................................................................... 15

1.3.3. Checkpoint File .......................................................................................................................... 15

1.3.4. All Other SSIS Objects ................................................................................................................ 15

1.4. SSRS ................................................................................................................................................... 15

1.5. SSAS................................................................................................................................................... 15 1.5.1. Hierarchies ................................................................................................................................. 16

# 1. Naming Conventions

## 1.1. Instances

instances will be named instances with the following naming conventions based on the databases hosted on the instance as follows:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

except for a standard DBA database which contains admin code to manage the instance.

The database engine on the backend of SSRS will be hosted on a dedicated instance. This instance will hold the SSRS databases and any jobs for managing/delivering reporting solutions.

COTS Internal: A database for off the shelf software which contains data which is for internal CBRE use only e.g. the database(s) underneath ServiceNow, VMWare, Altiris, etc.

COTS External: A database for off the shelf software which contains customer data which is used by CBRE account teams to manage accounts as well as by customers for managing their portfolios e.g. MRI, Harbor Flex, Yardi, etc.

Custom: A database for an application which has either been custom built for CBRE or is a COTS solution which has been customized to contain CBRE specific database objects. These databases contain customer data which is used by CBRE account teams to manage accounts as well as by customers for managing their portofolios e.g. ServiceInsight, CBRE 360, Atlas, GVAS, ValueTrack, etc.

## 1.2. Databases

**1.2.1. All Objects**

object will reflect the general or specific purpose the object is being created for. This naming convention is particularly important for schemas, tables, views, and columns. The reason for specifying upper Camel case is not only for clarity, but also speed of development on all reporting and analytic deliverables. The reporting environments we use such as SQL Server Reporting Services and SQL Server is Services automatically recognize and format upper Camel case names into human readable

format e.g. FirstName is translated to First Name. By using upper Camel case, we reduce the amount of reformatting of identifiers and labels in the production of every reporting deliverable.

Since our data models are published, upper Camel case is also readable by non-technical people so they can easily understand what we are capturing and how the data structures are laid out.

You are going to spend a lot of time typing object names. Object names can be up to 128 characters in length. While you should be descriptive in the names of your objects, you also don’t want to name something such that someone’s fingers are going to fall off when they have to type it. You should also avoid names that have very small differences from the name of another object as this leads to bugs which are extremely hard to find. However, abbreviating to an unintelligible mess is also not allowed.

Examples: FirstName, EmployeeID, PropertyName

The use of T-SQL or ODBC reserved words, which are documented in Books Online, are not allowed in object names as these are structural elements that the SQL Server parser uses to understand the text you submit as code directives. T-SQL keywords are a much larger set of words used by SQL Server. As SQL Server add features, it becomes more difficult to avoid T-SQL keywords. The list of keywords has expanded from a few hundred at SQL Server 4.21a to over ten thousand with SQL Server 2012. For example, contract and certificate are standard business terms and it is very likely objects already exist in databases with these names. When possible, T-SQL keywords should not be used for object names.

### 1.2.2. General Naming

based on US English. It is understood that schemas contain

multiple objects, tables contain multiple columns, and columns contain multiple rows.

The primary reason for object names to be singular is based in language semantics, any language’s semantics. Every word, in every language has a singular form. However, not every word has a plural form. Additionally, abbreviations and acronyms are singular and do not have plural forms.

Naming conventions are designed to provide unambiguous rules. Standards with numerous exceptions, open to interpretation, or subject to extensive knowledge of a language lead to chaos. Having a naming convention which dictates plural names leads to things like Fishs, Corns, Orcas, Gooses, Mgrs, TCPIPs, IDs, NOIs, UOMs, etc.

### 1.2.3. Database

purpose of the objects within the database. Database names will not contain data, db, database, or any other permutation as these do not convey any useful information, unless those terms are an actual part of application branding.

Not allowed: dbAtlas, db\_Atlas, AtlasDB, AtlasData

### 1.2.4. Filegroups

additional filegroups will reflect the name of the

database, an FG suffix, and a sequential number by default. If a filegroup is created for a special purpose on the database, that purpose will also be part of the name of the filegroup.

Examples: AtlasFG1, AtlasFG2, AtlasPropertyPartitionFG1, AtlastPropertyPartitionFG2,

AtlasPropertyDocFileStreamFG1

If a filegroup is used for dual purposes such as storing regular tables as well as partitioned tables, the name of the filegroup should reflect the default naming scheme.

### 1.2.5. Database Files

sequential number suffix. The first transaction log for a database will not have a suffix, but all additional transaction logs will contain a sequential number suffix.

The primary file for the database will have an extension of .mdf, secondary data files will have an extension of .ndf, and transaction logs will have an extension of .ldf.

Examples: Atlas.mdf, Atlas1.ndf, Atlas2.ndf, Atlas.ldf, Atlas1.ldf

Invalid Names: Atlas\_Data.mdf, Atlas\_Log.ldf

### 1.2.6. Schema

contained in the schema.

Examples: Marketing, Property, Admin, Lease

In some cases, we customize vendor software and include our one database objects inside the vendor database. All CBRE objects added to a vendor database will be maintained in one or more separate schemas which reflect the general subject of the objects in the schema. These schemas will be prefixed by CBRE.

Examples: CBREMarketing, CBREProperty, CBREAdmin, CBRELease

The data source views for cubes are all defined over a set of views. These views should be separated into one or more schemas. These schemas will be prefixed with Cube and the subject area the views belong to.

### 1.2.7. Table

the table.

Examples: Lease, Resident, Prospect

#### 1.2.7.1. Data Warehouse and Data Mart

which embed type logic into the table name such as DimProperty, DimLease, FactTransaction. This type of naming convention is not allowed as it creates confusion and inappropriately classifies the use of a given set of data. You analyze facts by dimensions. However, the same block of data is a dimension to one person and a fact to another person. As really good example of this is a Lease. Most people would immediately say that Lease is a dimension. Yes, to an accounting team rolling up payments by leases, the Lease is a dimension. However, a marketing group would treat leases as a fact when trying to determine what is available and a business group responsible for determining occupancy would treat a lease as a fact. Same data, completely different purposes between teams. If you were to prefix the name of the Lease table with Dim, everyone would assume that it is a dimension and always a dimension and so would assume that someone made a mistake when they see a table prefixed with Dim as the source of aggregates for a cube.

Data marts are purpose built extracts of a larger data warehouse. You generally predetermine the role of every table in a data mart when it is built based on what you choose to include. However, business users have a habit of asking applications and data to do things you hadn’t considered. In a data warehouse, you never predetermine the role that a piece of data will have as that limits the types of analysis which can be performed. Data simply is. It is our job to name it appropriately, store it, and make it available. It is up to the business to determine what role, fact or dimension, that a given piece of data plays in analysis.

### 1.2.8. Column

stored.

Examples: FirstName, StreetAddress, Gender

#### 1.2.8.1. System Generated IDs

s a system generated identifier will be designated with a suffix of ID. This is very

explicitly ID, not Id. ID is an abbreviation for Identification Document, not identifier or identification, which is why both characters are capitalized.

ID, PropertyID

#### 1.2.8.2. Boolean Columns

Choices such as True/False, Yes/No, and On/Off are generically referred to as indicators. All indicator type columns will end with the word Flag. All columns which are not an indicator type will follow the normal rules for naming columns.

Examples: ActiveFlag, OccupiedFlag

#### 1.2.8.3. Bitmask columns

columns which are internal application use only and will never be displayed to a user, used in a report, or extracted to another system. Bitmask columns will contain a suffix indicator of BitMask.

Examples: DayBitMask, LevelBitMask

#### 1.2.8.4. Date and Time Columns

column stores a time only, the name of the column will end in the word “Time”. If the column contains dates only or a date + time or a date + time + time zone, the column name will end in the word “Date”.

Examples: CreateDate, LeaseDate, LastModifiedTime, CurrencyDate

### 1.2.9. View

Some naming standards specify a prefix for a view and may also specify a prefix for a table name. Both of these conventions create a potential management issue where the development team may be required to violate the naming convention in order to accomplish a business requirement. You are allowed to modify data in a table as well as through a view. You might start off with a process reading/writing data to a table, but in order to accomplish a future requirement, you could replace the table with a view and not have to refactor all of the code. By forcing a naming convention on either views or tables, you can force the development team to obey the naming convention and incur a significant amount of work or violate the naming convention.

Just as our tables do not have a prefix convention, our views will not have a naming convention prefix.

Examples: CustomersByProperty, ProspectHistory, CustomerAccountBalanceByGLAccount

**1.2.10. User Defined Data Type**

### 1.2.11. Foreign key

oreign keys will be named with a prefix of fk\_ and describe the tables and columns involved in the

relationship. The name will be of the following form: fk\_<Parent Table>To<Child Table>On<Column Name>

Examples: fk\_LeaseToPropertyOnLeaseID, fk\_EmployeeToOrganizationOnManagerID

### 1.2.12. Primary key

the following form: pk\_<Table Name>

Examples: pk\_Building, pk\_Prospect, pk\_Lease

### 1.2.13. Index

exes which can be created and some of the options can be combined. Each type of index will have its own prefix, but all will start with a prefix of i followed by one or more type designations and an underscore. The primary key naming convention takes precedence over this convention.

Index names will be of the form <prefix><index type>\_<list of columns or description>. The suffix can be either a list of columns or a description whichever is more appropriate and conveys more meaning. When using a description, the description should be enough to explain the purpose of the index. When a list of columns is chosen, the list should contain enough columns in the name to understand the scope of the index, but does not need to exhaustively list all columns.

Some options are exclusive, while other options can be combined. For example, you can have a unique clustered index, but you can’t have an index that is both clustered and non-clustered. For those cases where you can apply multiple options to a single index, the precedence order listed below will dictate the order of the elements in the prefix.

Examples are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  | -clustered |  |  |
|  |  |  |  |
|  |  |  |  |
|  | -text | ft |  |
|  |  |  |  |
|  | – geometry |  |  |
|  | – Primary |  |  |
|  | – Path |  |  |
|  | – Property |  |  |
|  | – Value |  |  |
|  | - primary |  |  |
|  | - secondary |  |  |

named: iunf\_Market\_IsNotNull

### 1.2.14. Default

h tells the reader the purpose of the default. All defaults are required to be named and will never utilize the default system naming.

Examples: df\_CurrentDate, df\_ZeroAmount

### 1.2.15. Check constraint

y a descriptive name which tells the

reader the purpose of the constraint. All check constraints are required to be named and will never utilize the default system naming.

Examples: ck\_PriceGreaterThanZero, ck\_ValidEMailAddress

### 1.2.16. Unique constraint

nstraints will be named with a prefix of uc\_ following be a descriptive name which tells the

reader the purpose of the constraint. All unique constraints are required to be named and will never utilize the default system naming.

Examples: uc\_StateProvinceWithinCountry, uc\_CustomerName, uc\_AccountName

### 1.2.17. Stored procedure

|  |  |
| --- | --- |
|  |  |
|  | tored procedures used for application purposes |
|  |  |
|  |  |
|  | oad test stored procedures |
|  |  |
|  |  |

utp\_GetOrganizationHierarchy, ltp\_GetOrganizationHierarchy, sbap\_ProcessETL, rgcp\_ClassifiyUsers

Resource Governor is a special case. The classifier routine runs during the authentication process to an instance. Therefore, only one classifier stored procedure can be active on an instance at any time. The classifier stored procedure is installed in the master database.

### 1.2.18. Trigger

– DML, DDL, and logon. Triggers will have a prefix specific to their

purpose as shown in the following table.

|  |  |
| --- | --- |
|  |  |
|  | actions the trigger is applied to: **i**nsert, **u**pdate, and **d**elete. |
|  |  |
|  |  |

triggers will be followed by the table name. The prefix for DDL and logon triggers will

be followed by a descriptive name that captures the purpose of the trigger.

Examples: ti\_Property, tddl\_PreventObjectDrop, tl\_LimitMultipleConnections

### 1.2.19. Function

ns will be prefixed with fn\_ followed by a descriptive name. While you can have four different

styles of function: scalar, table valued, multi-statement table valued, and CLR where functions are used and what they are applied to do not change. Therefore, the naming convention will not distinguish between the different types of function.

### 1.2.20. XML Schema Collection

will have a prefix of xsc\_ followed by a descriptive name that captures the purpose of the schema collection.

Examples: xsc\_ValidateETLInitializationMessage, xsc\_NotifyCompletionOfDataLoad

### 1.2.21. Partition Function

function will be applied to in the following form pfn\_<schema>\_<table or indexed view name>

Examples: pfn\_Marketing\_Prospect, pfn\_Accounting\_Payments

### 1.2.22. Partition Scheme

tition scheme will be applied to in the following form: ps\_<schema>\_<table or indexed view name>.

Examples: ps\_Marketing\_Prospect, ps\_Accounting\_Payments

### 1.2.23. Service Broker

entation is referred to as an application, because it requires multiple objects to be created before anything useful can be done. All Service Broker objects will utilize a naming convention designed to group all of the components to a specific application as follows: sb<object type prefix>\_<application name>\_<object name>.

**1.2.23.1. Message Type**

Examples: sbmt\_Atlas\_LoadProperties, sbmt\_Atlas\_LogRunStats

#### 1.2.23.2. Contract

along with a descriptive name.

Examples: sbct\_Atlas\_LoadProperties, sbct\_Atlas\_LogRunStats

**1.2.23.3. Queue**

Example: sbq\_Atlas\_LoadPropertiesInbound, sbq\_Atlas\_LoadPropertiesAcknowledgement

#### 1.2.23.4. Service

es are used to wire contracts to queues. Since a service can be used to interact with remote

routes defined for a Service Broker application, services will follow a global naming scheme outside of the standard rules for naming database objects. Since this naming convention violates the standard QL Server rules for identifiers, it must be enclosed in square brackets. The naming convention for a service will be:

[//CBRE.com/<application>/sbsr\_<descriptive name>To<queue description>]

Example: [//CBRE.com/Atlas/sbsr\_SendRunStatsToAcknowledgementQueue]

This naming convention also creates issues for file names in Windows. Scripts for services will follow the file naming rules for source control objects as outlined earlier in this document except the square brackets will be omitted from the file name and the forward slashes will be replaced with a single dash.

#### 1.2.23.5. Endpoint

processing. A single Service Broker application will not have multiple endpoints as such the endpoint name will simply be the name of the Service Broker application.

Examples: Atlas, GlobalInsight

#### 1.2.23.6. Route

sbrt\_<application name>\_<description>

Example: sbrt\_Atlas\_OffloadLogging

### 1.2.24. Linked Server

#### 1.2.24.1. Replication

##### 1.2.24.1.1. Publication

ame which explains what the publication is

doing. The replication method does not need to be part of the name. In many cases, the default name should be sufficient, but should be modified when the default name does not accurately capture what the publication is for.

Example: ServiceInsight Read Only Tables, Atlas Properties

The file naming convention for source control will be of the following form:

<publication name>.<replication method>.Publication.sql

Example: Atlas Properties.Transactional.Publication.sql

##### 1.2.24.1.2. Subscription

as well as what the subscription is doing. The replication method does not need to be part of the name. In many cases, the default name should be sufficient, but should be modified when the default name does not accurately capture what the publication is for.

Example: ServiceInsight Read Only Tables From USCDCMKS82

The file naming convention for source control will be of the following form:

<subscription name>.<replication method>.Publication.sql

#### 1.2.24.2. SQL Server Agent

Names will be constructed such that related objects will group together alphabetically

#### 1.2.24.3. Audit

style, human readable names which explain the purpose of the object.

Examples: Security Violation Event Logging, Use of Generic Account, Use of Elevated Permissions to Access Payroll Data

#### 1.2.24.4. Policy Management

##### 1.2.24.4.1. Policy

ugh to understand the purpose of the policy.

Example: CheckTablesHavePrimaryKey, FullBackupInLast7Days

##### 1.2.24.4.2. Condition

enough to understand the purpose of the condition.

Example: CheckTranLogBackupInLast4HoursForFullRecoveryDatabases

#### 1.2.24.5. Extended Event Session

descriptive enough to understand what the session is doing.

Example: Sample20PercentQueryExecutionToFile

## 1.3. SSIS

### 1.3.1. Package Name

This will consist of a “dotted” notation scheme which encompasses subject area and purpose of the package. The solution provides the top level portion of the naming scheme. For example if we were to build a series of SSIS packages to load the data from Atlas into our data warehouse, the solution would be named Atlas and within the Atlas solution we would have a set of packages with the following naming convention:

<subject area>.dtsx

Data Flow Packages

<subject area>.<destination table>.dtsx

Example: Lease.Lease.dtsx, Property.Building.dtsx, Property.dtsx

### 1.3.2. Configuration File

Configuration file names will correspond to the package name and contain an additional element to designate the environment the configuration file should belong to with the following form:

<subject area>.<purpose>.<Environment>.dtsconfig

Example: Lease.LoadLease.Dev.dtsconfig, Property.LoadBuilding.Prod.dtsconfig

### 1.3.3. Checkpoint File

area>.<purpose>.ssischkpt

Example: Lease.LoadLease.ssischkpt, Property.LoadBuilding.ssischkpt

### 1.3.4. All Other SSIS Objects

and abbreviations will be kept to a minimum. Each object is required to have a descriptive name and will not use the default object names or simply repeat the name of another object with an appended sequence number.

## 1.4. SSRS

e separated by a space and abbreviations will be kept to a minimum. Default names for objects will not be used or simply repeat the name of another object with an appended sequence number.

## 1.5. SSAS

ect, because business users directly interact with the structural elements of a cube or mining model. All SSAS objects will have descriptive, English style names. Each word will be separated by a space and abbreviations will be kept to a minimum. Default names for objects will not be used or simply repeat the name of another object with an appended sequence number. “Techno-terms” will not be used in any object name visible to a business user. For example, Dim and Fact will not be prepended or appended to any SSAS object name. confusing and irrelevant to their work.

*Improper Naming:*

Fact Financial, Dim Bus Unit, Dim Date

*Proper Naming:* Financial, Business Unit, Date

### 1.5.1. Hierarchies

name of the hierarchy with By.

Example: By Organization Unit, By Geography, By Fiscal Year